

Installing K3s with Longhorn and USB storage on Raspberry Pi

April 10, 2021

Just sharing this for my future reference as well.

1. Install Ubuntu

Raspberry Pi OS (previously Raspbian) has not released its 64-bit build yet. I also do not want to be bothered by the [iptables work around for Traefik in Raspberry Pi OS](#).

1. Download the 64-bit image [here](#)
2. Restore the image to your SD card/s.
3. Unlike in Raspberry Pi OS, do not need to create an ssh file in /boot to enable ssh.
4. append:

```
cgroup_memory=1 cgroup_enable=memory
```

to your /boot/firmware/cmdline.txt file.

5. If you forgot the previous step, just ssh in to your Pi, modify said file and reboot.

2. Install [K3s](#)

1. On your master node: If your master node is behind a router (i.e. you are port forwarding),

```
curl -sfL https://get.k3s.io | INSTALL_K3S_EXEC="--tls-san [your router IP]" sh -s -
```

Otherwise,

```
curl -sfL https://get.k3s.io | sh -s -
```

2. Check master K3s status with:

```
systemctl status k3s.service
```

No error? Good.

3. Copy the k3s kube config file to your client (probably your desktop pc). The file should be located at /etc/rancher/k3s/k3s.yaml on master node. Copy it to your client's .kube/config
4. Modify .kube/config file,
change clusters[0].cluster.server value
from <https://127.0.0.1:6443> to whatever your master node ip is

(if you are port forwarding, your router's IP)

example: <https://192.168.1.1:6443>.

5. Check connection from client with:

```
kubectl get node
```

Expected result:

NAME	STATUS	ROLES	AGE	VERSION
master	Ready	control-plane,master	10m	v1.20.5+k3s1

6. Get the K3s token on your master with:

```
sudo cat /var/lib/rancher/k3s/server/node-token
```

7. On your worker nodes:

```
curl -sfL https://get.k3s.io | K3S_URL=https://[master IP]:6443 K3S_TOKEN="[K3s token]" sh -
```

8. Check worker K3s status with:

```
systemctl status k3s-agent.service
```

No error? Good.

9. Check connection from client with:

```
kubectl get node
```

Expected result:

NAME	STATUS	ROLES	AGE	VERSION
master	Ready	control-plane,master	20m	v1.20.5+k3s1
worker	Ready	<none>	10m	v1.20.5+k3s1

10. Reference: <https://rancher.com/docs/k3s/latest/en/>

3. Optional: Install [Web UI \(Dashboard\)](#)

1. On your client, run:

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0/aio/deploy/recommended.yaml
```

2. Create admin user

```
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: ServiceAccount
metadata:
```

```
name: admin-user
namespace: kubernetes-dashboard
EOF
```

3. Give cluster admin role to admin user

```
cat <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin-user
  roleRef:
    apiGroup: rbac.authorization.k8s.io
    kind: ClusterRole
    name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kubernetes-dashboard
EOF
```

4. Get bearer token:

```
kubectl -n kubernetes-dashboard describe secret admin-
user-token | grep ^token
```

5. Run proxy from client

```
kubectl proxy
```

6. On your browser,
open <http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/>. Enter the bearer token you have obtained.
7. Reference: <https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/>

4. Optional: Install [Longhorn](#)

Installing longhorn will allow you to have dynamic provisioning for persistent volume claims.

1. On your client, run:

```
kubectl apply -f
https://raw.githubusercontent.com/longhorn/longhorn/master/deploy/longhorn.yaml
```

2. Wait a little while, you may check the status of the pods via the web ui or via:

```
kubectl -n longhorn-system get pods
```

3. Test! Create a PVC. On your client, run:

```
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: test-pvc
spec:
  accessModes:
  - ReadWriteOnce
  storageClassName: longhorn
resources:
  requests:
  storage: 1Gi
EOF
```

4. Test! Create a Pod that uses the PVC. On your client, run:

```
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: volume-test
spec:
  containers:
  - name: volume-test
    image: nginx:stable-alpine
    imagePullPolicy: IfNotPresent
    volumeMounts:
    - name: test-pvc
      mountPath: /data
  ports:
  - containerPort: 80
  volumes:
  - name: test-pvc
    persistentVolumeClaim:
      claimName: test-pvc
EOF
```

5. Check status, on your client, run:

```
kubectl get pods
```

If all goes well, you should see the pod running well:

```
NAME READY STATUS RESTARTS AGE
volume-test 1/1 Running 0 29s
```

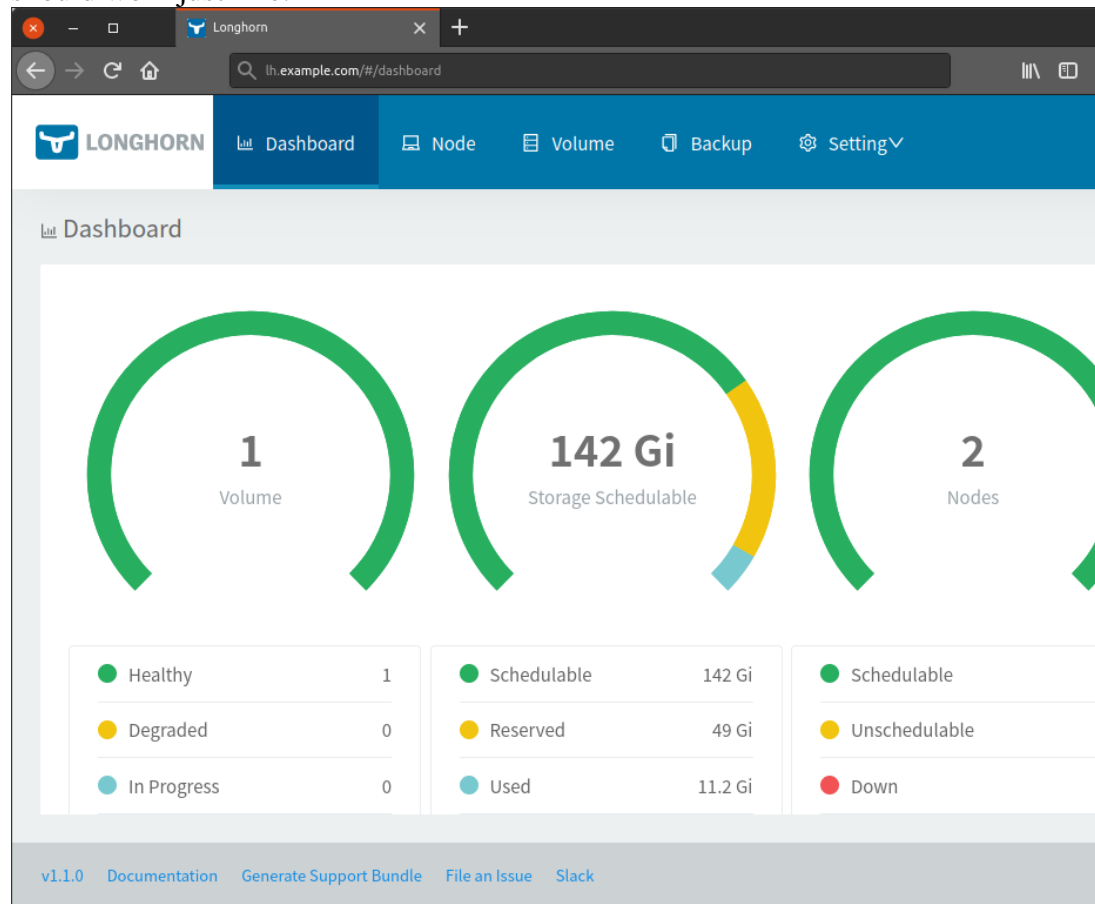
6. Clean up:

```
kubectl delete pvc/test-pvc pod/volume-test
```

7. Optionally, you can open up Longhorn UI Ingress with:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  namespace: longhorn-system
  name: longhorn-ingress
  annotations:
    kubernetes.io/ingress.class: "traefik"
spec:
  rules:
  - host: longhorn.example.com
    http:
      paths:
      - path: /
        pathType: Prefix
      backend:
        service:
          name: longhorn-frontend
          port:
            number: 80
```

8. If you do not own a domain, just mock it with `/etc/hosts` and it should work just fine.



- 9.

- 10.
11. If you are not serious about storage replicas, you may want to change it to 1 replica, default is 3 replicas.

```
kubectl -n longhorn-system edit cm/longhorn-storageclass
```

change numberOfReplicas to 1

12. Reference: <https://rancher.com/docs/k3s/latest/en/storage/>

5. Optional: Use USB Storage Device for Longhorn

1. Connect your USB storage device to one of your nodes.
2. `ssh` into that node.
3. Create mount point

```
sudo mkdir /media/storage
```

4. Get the PARTUUID with:

```
sudo blkid
```

You should have something like:

```
/dev/sda1: LABEL="mystorage" UUID="[device uuid]"  
TYPE="ext4" PARTUUID="[partition uuid]"
```

5. modify your `/etc/fstab` file by adding the line (note: my external device file system is ext4):

```
PARTUUID=[partition uuid] /media/storage ext4  
defaults,noatime,nodiratime 0 2
```

6. You can test if it mounted correctly with:

```
sudo mount -a
```

7. K3s will run as root so it will have full access to the device. You can optionally create a group for the directory and add yourself to give yourself access to the files:

```
sudo groupadd [group name]  
sudo usermod -aG [group name] [your username]  
sudo chown -R :[group name] /media/storage  
# set the gid bit so all files/directories created will have the same group as the  
sudo chmod g+s /media/storage
```

8. Add the storage to longhorn, access the longhorn UI, go to nodes and select "Edit node and disks", click "Add Disk". Fill up the details and click save.

+ New Disk Tag

Storage Available	Storage Scheduled	Storage Max
0Gi	0Gi	0Gi

Name

usb-storage

Path

/media/storage/longhorn/

Storage Reserved


32Gi

Scheduling

☐ Enable ☒ Disable

Eviction Requested

☐ True ☒ False



Apply

Cancel

9. And you are good to go!